Intel® VTune™ Amplifier XE 2017 Update 2 **What's New**

Intel® VTune™ Amplifier XE 2017 performance profiler

A performance profiler for serial and parallel performance analysis. Overview, training, support.

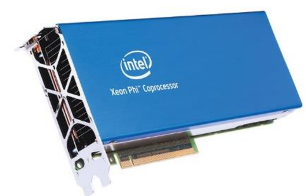
New for the 2017 Update 2 release!

As compared to 2017 Update 1 release

Support for cross-OS analysis to all license types

Starting with **2017 update 2** of VTune Amplifier ***XE*** any developer with a currently supported license will be able to use the tool on all supported operating systems. The default download will remain the same, but additional operating systems will be available for optional download from the Intel Registration Center.
See more details and FAQ: https://software.intel.com/en-us/articles/intel-vtune-amplifier-intel-advisor-and-intel-inspector-now-include-cross-os-support

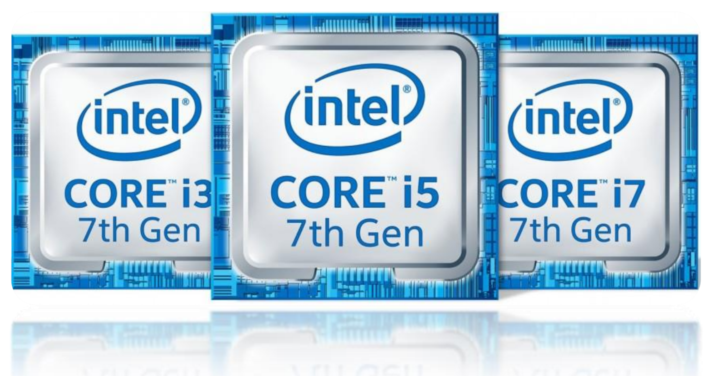
Support for Intel® Xeon Phi™ coprocessor codenamed Knights Landing



Intel® Xeon Phi™ coprocessor codenamed Knights Landing is supported by VTune Amplifier **from Linux*** OS host with hardware event based sampling analysis including **General Exploration**, **Memory Access** and **HPC Performance Characterization**. Support is enabled for both native and offload applications. Installation automatically configures SEP drivers and collector on the card.
For more details on analysis configuration see the Intel® Xeon Phi™ Coprocessor Analysis Workflow product help article.

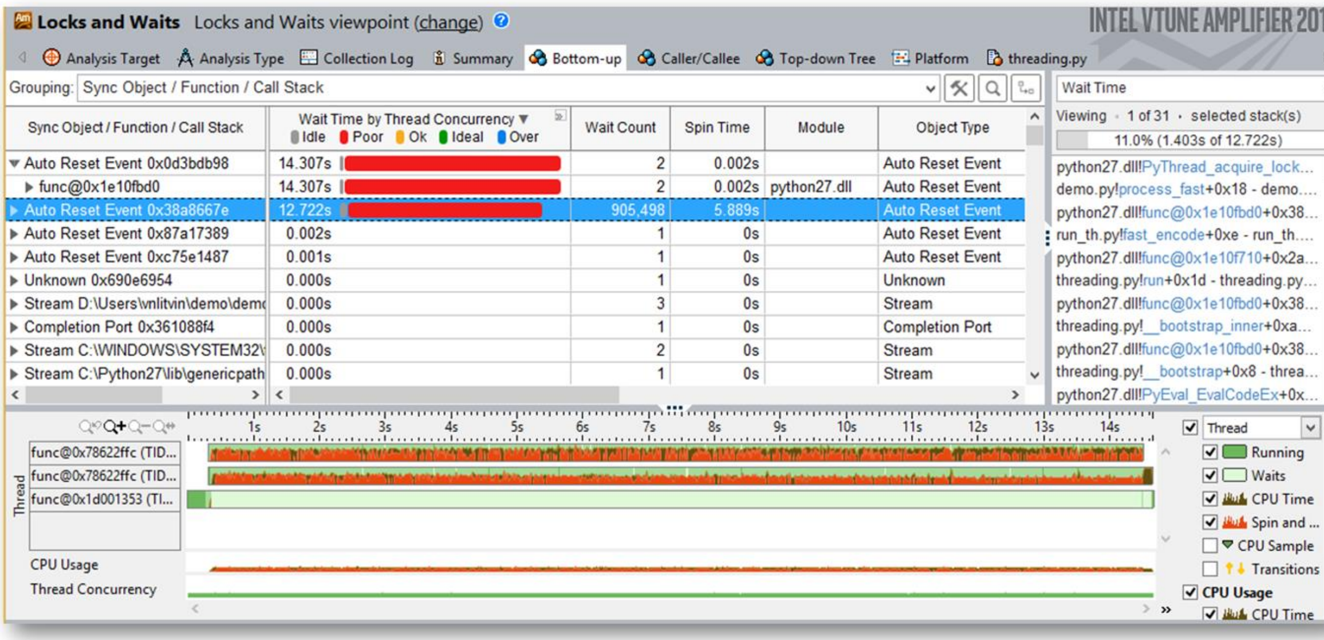
New processors support

Intel® Atom™ Processor E3900 Series (formerly **Apollo Lake)**, Intel® Atom™ Processor C3000 Product Family (formerly **Denverton)**, and the 7th Generation Intel® Core™ Processor Family (formerly **Kaby Lake)** are now publicly supported by VTune Amplifier with all pre-defined hardware event based analysis.



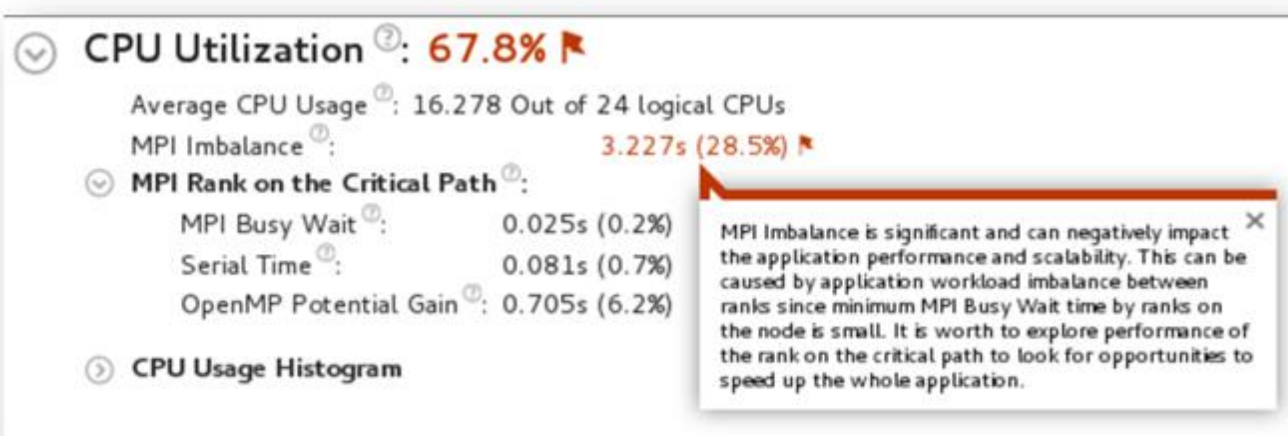
Locks and Waits analysis for Python

Locks and Waits analysis now tunes threaded performance of mixed Python* and native code. View Sync Objects in grid, see Python frames in Call Stack and define which sync objects are "the GIL" (Global Interpreter Lock), either by wait count or by callstack. Drill down to Python source to explore thread synchronization issues at code level.
See how to configure the analysis in the Python* Code Analysis product help article.



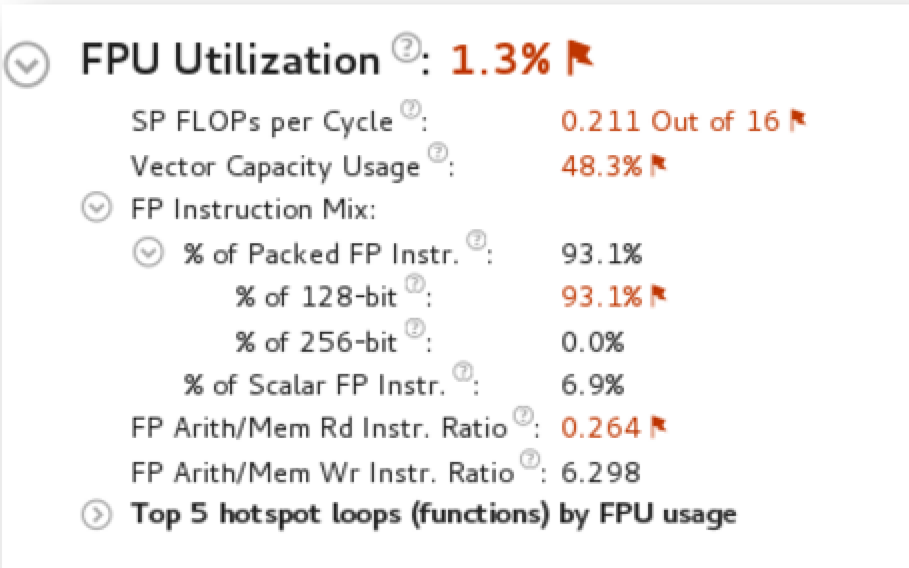
HPC Performance Characterization analysis improvements

For MPI applications, review the **MPI Imbalance metric** that shows the CPU time spent by ranks spinning in waits on communication operations, normalized by number of ranks on the profiling node. The metric issue detection description generation is based on minimal MPI Busy Wait time by ranks. If the minimal MPI Busy wait time by ranks is not significant, then the rank on with the minimal time most likely lies on the critical path of application execution. In this case, review the CPU utilization metrics by this rank.

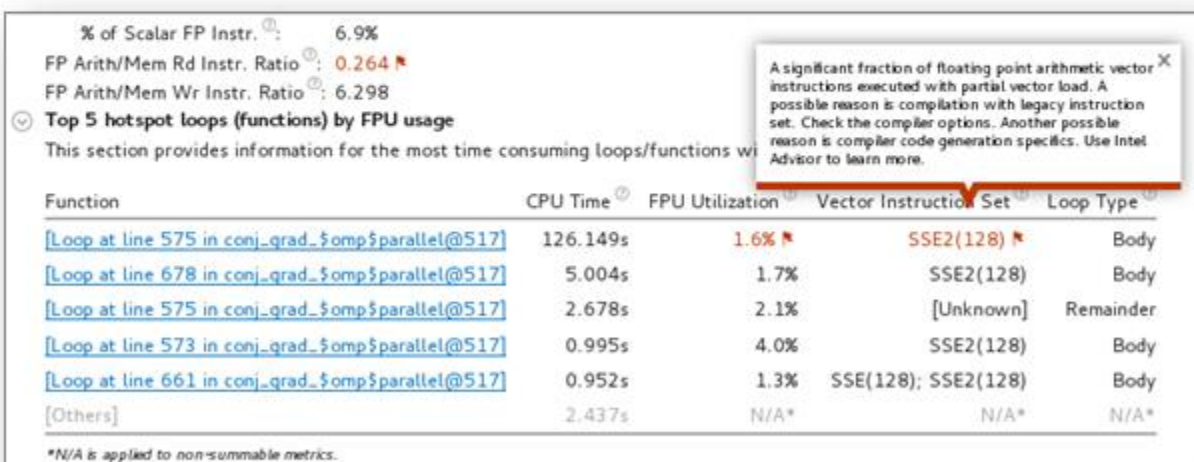


Increased detail and structure for vector efficiency metrics based on FLOP counters in the **FPU Utilization** section will help to learn the reason for low utilization connected with poor vector code generation.
Relevant metrics might include:

- **Vector Capacity Usage**
- **FP Instruction Mix**
- **FP Arithmetic Instructions per Memory Read or Write**
- **SP FLOPs per Cycle**, which could indicate memory bandwidth bound code

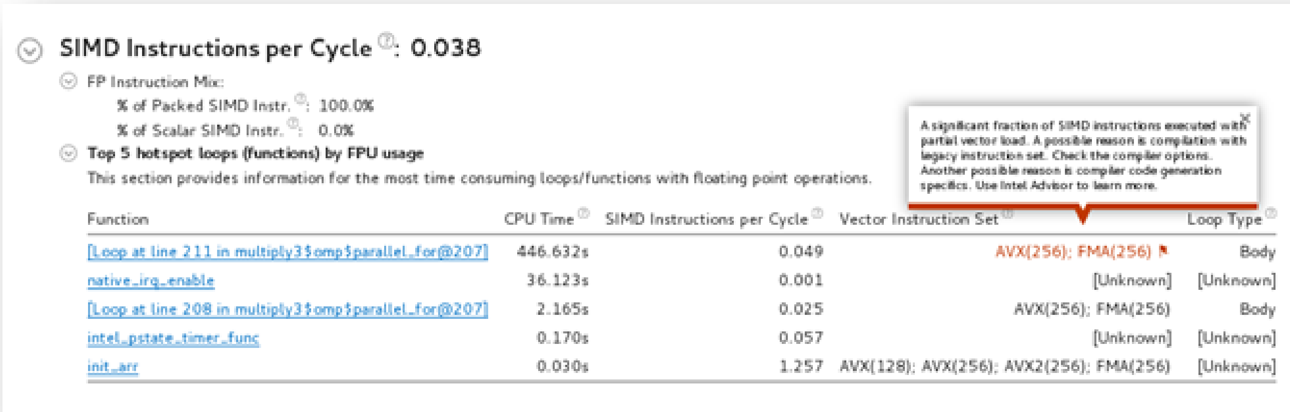


The **Top Loops/Functions with FPU Usage** by CPU Time table shows the top functions that contain floating point operations sorted by CPU time.
The **FPU Utilization** column provides issue descriptions based on whether a loop/function is bandwidth bound, vector instruction set, scalar, or vector.



For Intel® Xeon Phi™ processors code named Knights Landing, the following FPU metrics are available instead of FLOP counters:

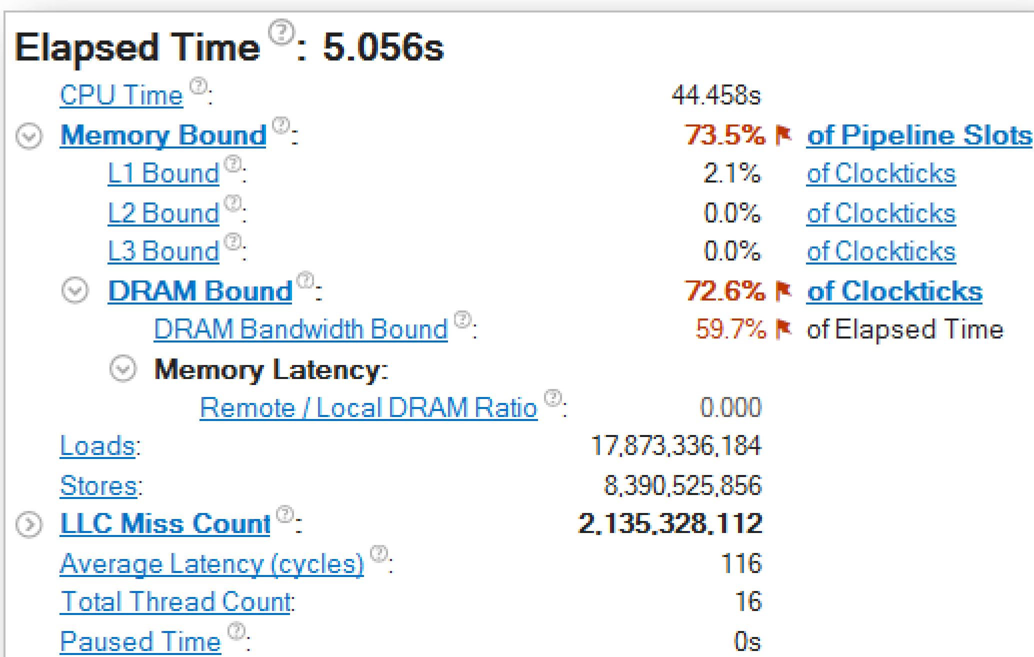
- SIMD Instructions per Cycle
- Fraction of packed SIMD instructions versus scalar SIMD Instructions per cycle
- Vector instructions for loops set based on static analysis



For more details see the HPC Performance Characterization analysis help article.

DRAM Bandwidth Bound metric

A high **DRAM Bandwidth Bound** metric value indicates that your system spent much time heavily utilizing the DRAM bandwidth. The calculation of this metric relies on the accurate maximum system DRAM bandwidth measurement and depends on the number of sockets on your system.

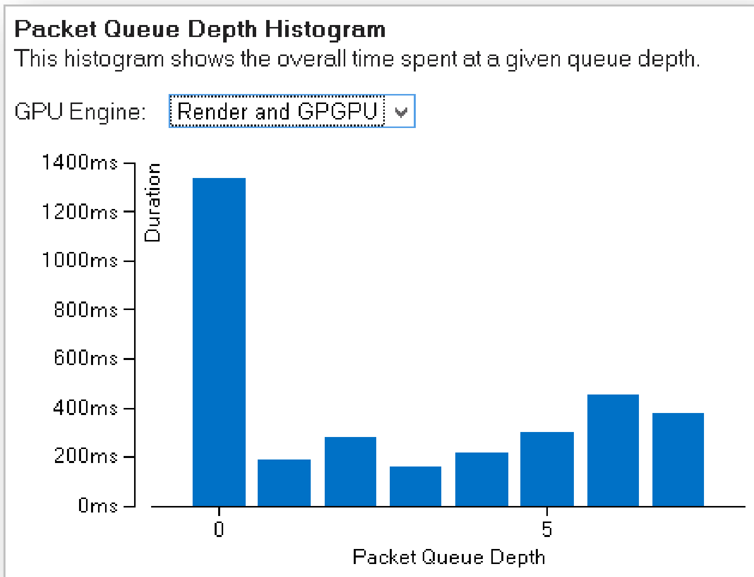


For more details see the Summary - Memory Usage product help article.

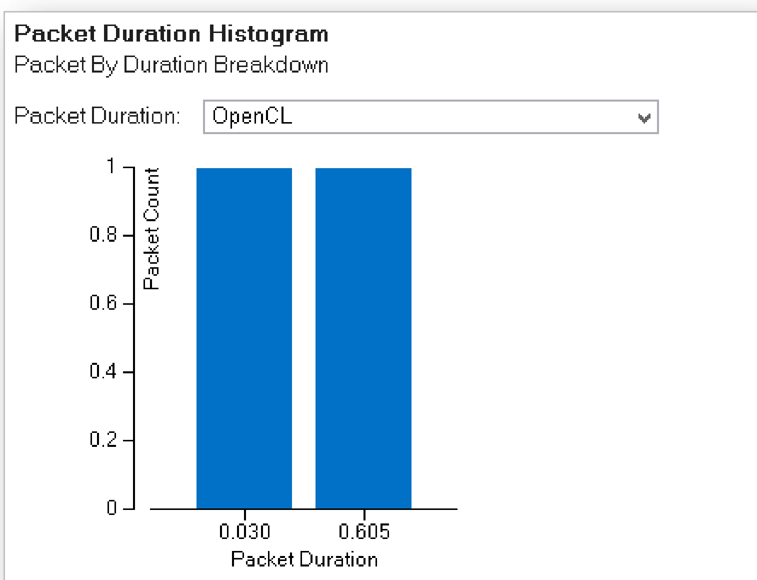
GPU Hotspots: Packet Queue Depth and Packet Duration histograms

In the Summary window of the **GPU Hotspots** viewpoint analyze the GPU Usage section to identify whether the GPU was properly utilized.

Explore the **Packet Queue Depth Histogram** to estimate the GPU software queue depth per GPU engine during the target run. Ideally, your goal is an effective GPU engine utilization with evenly loaded queues and minimal duration for the zero queue depth.



For a high-level view of the DMA packet execution during the target run, review the **Packet Duration Histogram.** Select a required packet type from the drop-down menu and identify how effectively these packets were executed on the GPU. Having high Packet Count values for the minimal duration is optimal.

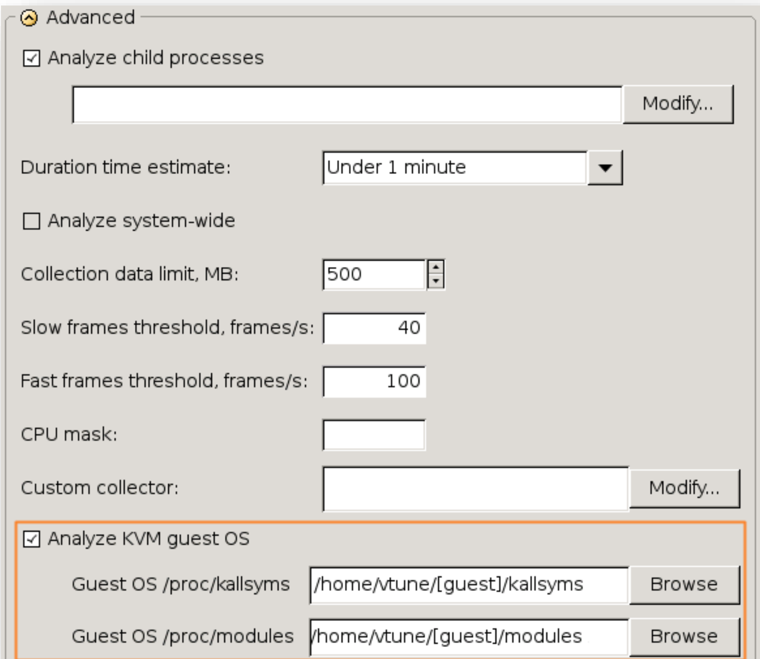


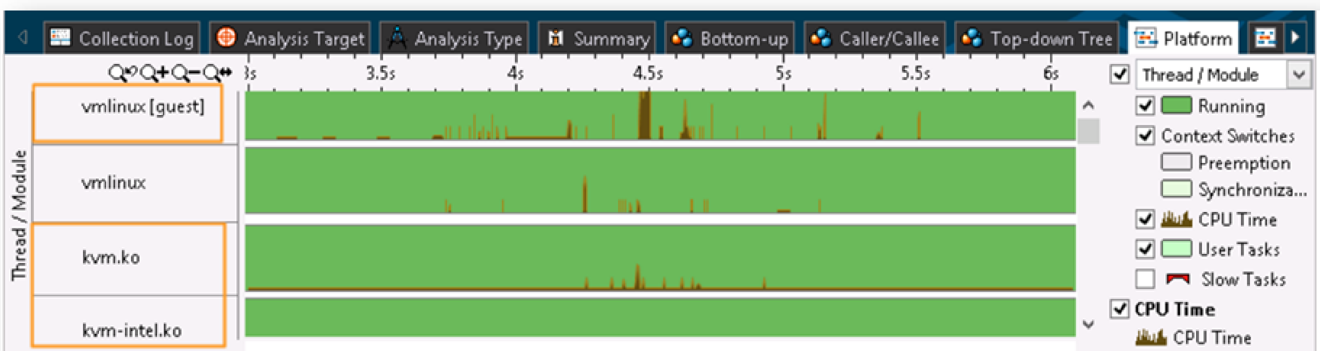
For more details see the https://software.intel.com/en-us/node/628123#GPU_UTILIZATION help article

KVM Guest OS Profiling

If you are a system developer and interested in the performance analysis of a guest Linux* system, use the VTune Amplifier for performance analysis of the guest Linux* OS via Kernel-based Virtual Machine (KVM) from the host system. Depending on your analysis target, you may choose any of the following usage models for KVM guest OS profiling:

- Guest system and KVM modules analysis
- Application analysis on the KVM guest system





For more details see the KVM Guest OS Profiling product help article.

Legal Information